УДК 598.3

О.А. Змеев, А.Н. Моисеев

ШАБЛОН ДИАГРАММЫ КОМПОНЕНТОВ ИНФОРМАЦИОННОЙ СИСТЕМЫ КОРПОРАТИВНОГО УРОВНЯ

В работе рассмотрен начальный этап проектирования информационной системы корпоративного уровня – создание пакетов приложения как семантически связанных групп объектов создаваемой программной системы.

В последнее время в нашей стране наблюдается устойчивый интерес к компьютерным интегрированным системам, способным обеспечить эффективное управление предприятием. Разработка и внедрение информационных систем управления предприятием, или корпоративных информационных систем (КИС), связана с определенными сложностями, возникающими уже на этапе начального проектирования архитектуры системы. Для объектно-ориентированных информационных систем рекомендуется использовать многоуровневую архитектуру, которая подразумевает распределение обязанностей, выполняемых объектами классической трехуровневой архитектуры. В процессе декомпозиции уровня предметной области трехуровневая архитектура превращается в многоуровневую без ярко выраженных границ уровня предметной области.

Можно подвергать дальнейшей декомпозиции существующие уровни и добавлять в систему дополнительные уровни. В качестве аргументов по использованию этого подхода при разработке КИС можно привести следующие обоснования: логика приложения представляется в виде изолированных компонентов, которые можно легко менять или использовать в других системах: различные уровни приложения можно распределить по разным компьютерам и/или нескольким процессам, что позволяет повысить производительность и улучшить координацию и совместное использование информации; разработку отдельных уровней можно поручить специализированным группам разработчиков, при этом поддерживается высокий уровень специализации профессионалов и возможность параллельной разработки всех уровней приложения. В статье рассмотрены типичные подходы к организации подобной архитектуры.

Некоторые особенности разработки КИС

Современные технологии менеджмента неразрывно связаны с понятием информационной системы корпоративного уровня (КИС). Для реализации таких систем уже недостаточно классических технологий программирования. Для современных проектировщиков предлагается широкий спектр инструментальных средств, позволяющих разрабатывать и реализовывать в виде исполняемых модулей системы этого класса.

Важное место среди них занимают CASE-средства. Под этим классом инструментария разработчика подразумеваются системы, позволяющие проектировать систему «с нуля», т.е. начиная с ее представления в виде модели. Далее процесс разработки принимает нисходящую направленность – крупные блоки дробятся, к полученным атомарным операциям применяются шаблоны, проектируются база данных, интерфейсные объекты и т.д. На финальном этапе формируются исполняемые модули.

Очень большой эффект на производительность разработчиков, задействованных в анализе и проектировании КИС, при этом оказывают использование объектно-ориентированных методов, унификация и специфицирование самого процесса разработки и проектирования. Эти технологии реализуются путем поддержки некоторого языка моделирования.

Наиболее распространенным и получившим международное признание разработчиков является язык объектного моделирования систем UML, представляющий собой мощное средство создания артефактов (документации), необходимых для проектирования и разработки программных средств. Богатый набор инструментов языка UML позволяет взглянуть на разрабатываемую систему с различных точек зрения. Причем переход от представления общих схем и диаграмм к описанию внутреннего устройства укрупненных объектов является достаточно естественным для этого языка. К тому же язык поддерживает большую библиотеку стереотипов (стандартных действий), а также создание собственных шаблонов, механизмов, каркасов и стереотипов.

Как видим, в руках современных разработчиков программных средств имеется достаточно мощный формальный инструмент – язык моделирования. Естественным для любого формального языка является его расширение на ту или иную предметную область. Причем это расширение достигается созданием шаблонов (типичных решений типичных проблем) и объединением их в соответствующие библиотеки. Достаточно вспомнить эволюцию какого-нибудь языка программирования на рынке инструментальных средств, когда не только компания-разработчик, но и «рядовые программисты» занимаются созданием достаточно мощных библиотек стандартных процедур, функций, объектов и т.п. При этом библиотеки различных разработчиков зачастую решают задачи в одной и той же прикладной области, создавая здоровую конкуренцию. Обычно только практический опыт использования этих библиотек позволяет сделать вывод о качестве той или иной разработки.

UML не является исключением из этого правила. Многие проектировщики систем в процессе работы с этим языком компонуют свои шаблоны в различных прикладных областях [1]. В настоящей работе предлагается несколько шаблонов (стандартных решений) проектирования архитектуры информационной системы корпоративного уровня.

В чем отличительная черта данных программных систем от других? Дело в том, что именно для информационных систем как нигде больше очень значимыми являются задачи сохранения информации, поиска, каскадных воздействий и других операций в хранилище данных, а также задачи взаимодействия с другими системами. Кроме того, чтобы систематизировать процесс разработки ИС, необходимо еще на этапе проектирования определить основные семантически связанные группы сущностей и объединить их в так называемые пакеты. Это тем более актуально для систем корпоративного уровня. В этом случае обязательно решаются вопросы централизованного доступа к хранимым данным, а это

почти всегда (по крайней мере, для эффективного решения задач) приводит к физическому разделению объектов программы – реализации системы на основе технологий архитектуры «клиент/сервер».

Таким образом, задача унификации процесса проектирования и определения основных шаблонов, предназначенных для разработки корпоративных информационных систем корпоративного уровня, является актуальной.

Итак, как уже говорилось, процесс проектирования КИС следует начать с определения пакетов – основных семантически связанных групп объектов, разделение на которые может быть использовано при реализации в архитектуре «клиент/сервер». Какие же основные группы объектов можно определить для ИСДО? На рис. 1 изображен шаблон классической модели пакетов программной системы [2].

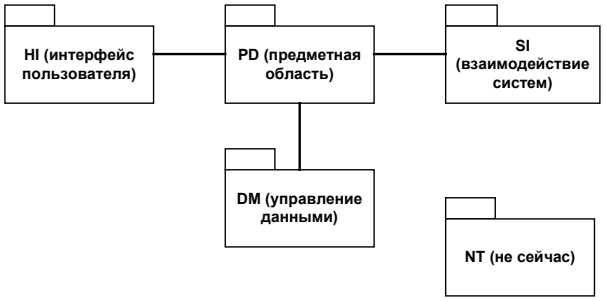

Рис.1. Основные пакеты в классической модели

Как показано на рисунке, таких пакетов пять:

– HI (Human Interface, интерфейс пользователя) – объекты, предназначенные для отображения, ввода и вывода информации на печать (окна, отчеты и объекты, связанные с ними);

– PD (Problem Domain, предметная область) – объекты, отражающие в программе объекты реального мира;

– SI (System Interface, взаимодействие систем) – объекты, предназначенные для связи с другими программными системами;

– DM (Data Management, управление данными) – объекты, отвечающие за доступ к хранимым данным и выполняющие все необходимые операции над ними;

– NT (Not This Time, не сейчас) – объекты, находящиеся вне функциональных или временных рамок программы.

Этот шаблон предлагает общее решение для любых программных средств, причем разделение объектов на пакеты не означает обязательного физического разделения. Более того, авторы не отрицают возможности существования объекта, расположенного сразу в нескольких пакетах. Практическое применение этого шаблона для решения конкретных задач позволяет получить типовые решения типичных проблем в контексте решаемой задачи.

Конкретизируем задачу: здесь и в дальнейшем речь пойдет только об информационных системах корпоративного уровня. Это, в частности, предполагает, что в качестве хранилища данных будет использоваться соответствующая база данных. Кроме того, система реализуется на основе технологии «клиент/сервер», а следовательно, объекты различных пакетов будут разделены физически – они будут находиться в разных исполняемых модулях. Рассмотрим, где же пройдет граница разделения пакетов. Согласно основным принципам технологии «клиент/сервер» (от которых мы в будущем все-таки отступим) объекты, реализующие представление данных (интерфейс пользователя), выносятся в отдельный программный модуль – «Клиент». Объекты оставшихся трех пакетов (PD, SI, DM, так как пакет NT вообще не реализуется) помещаются в программный модуль «Сервер», через который клиенты и получают доступ к информации. Такой шаблон представлен на рис. 2.

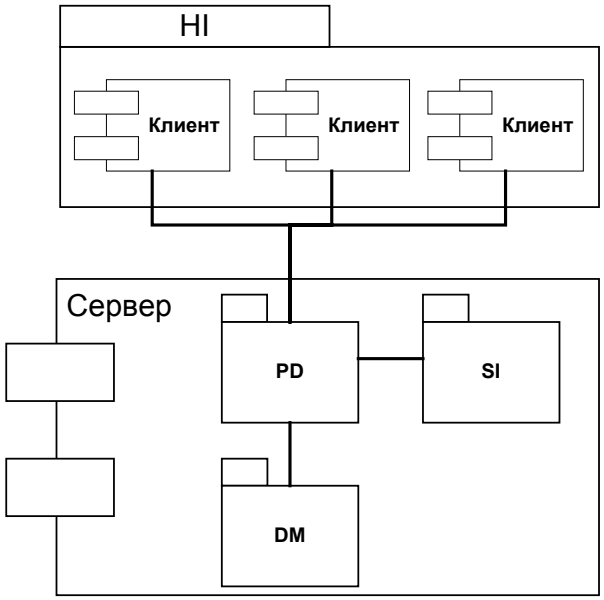

Рис. 2. Шаблон пакетов в двухуровневой архитектуре «клиент/сервер»

Современные условия эксплуатации информационных систем предполагают не только централизованный доступ к данным программ одной предметной области, но и централизацию доступа в пределах всего предприятия (корпоративный уровень). А это означает, что в пределах КИС может существовать несколько пакетов PD для каждой предметной области. Кроме того, некоторые объекты могут дублироваться в различных пакетах. Таким образом, логично выделить пакеты PD с базовыми объектами для групп предметных областей (возможно, такой пакет будет один) в отдельные программные модули, а оставшиеся в пакетах предметных областей объекты разместить в других программных модулях (рис. 3). Такой каркас программной системы представляет ее в виде многоуровневого сервера архитектуры «клиент/сервер».

Естественным при этом является вопрос: а что произойдет с пакетом (пакетами) DM? Представим, что вы используете во всех программах один и тот же формат баз данных (не только одна СУБД, но и одинаковые правила построения структуры). Тогда, естественно, все операции с информацией в этих базах данных будут происходить единообразно.

А это означает, что нет необходимости реализации нескольких серверов DM. Тем более что некоторые данные будут общими для различных PD. Таким образом, можно сделать вывод о необходимости реализации только одного пакета-сервера DM.

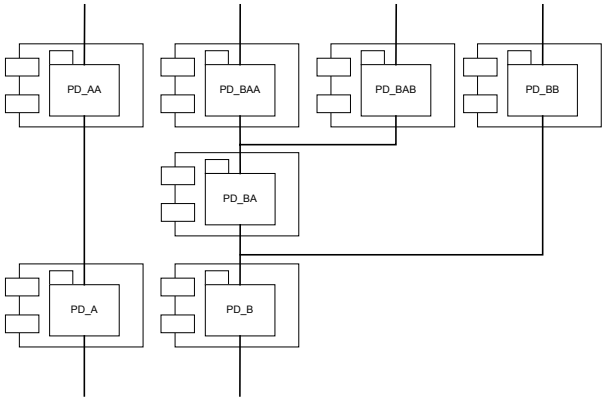

Рис. 3. Возможная схема
многоуровневого сервера предметных областей

Наконец, необходимо решить вопрос о пакетах SI. Здесь вывод о разделении объектов можно сделать на основе изучения назначения объектов этих пакетов. Их основная функция (рис.1) – поддержка интерфейса с другими системами. Какие это могут быть системы? Можно выделить основные виды таких систем:

– системы, реализованные другими разработчиками;

– ранее разработанные системы;

– системы, по каким-либо причинам реализованные на основе других моделей данных, не имеющие собственной базы данных, и т.п.

Отличительная черта таких систем – несовместимость формата базы данных с принятым в нашей системе. Таким образом, можно сделать вывод, что для обмена данными с каждой из таких систем (возможно, классов систем) необходимо реализовывать свой пакет SI, даже если такая связь осуществляется для программы с единственным пакетом PD и единственным пакетом HI. Вообще говоря, пакеты SI не обязательно присутствуют в каждом приложении.

Даже если программа должна осуществлять взаимодействие с другими системами, не всегда следует включать пакеты SI в общую диаграмму пакетов проекта. Это является темой отдельного обсуждения.

Общий шаблон диаграммы пакетов информационной системы корпоративного уровня представлен на рис. 4.

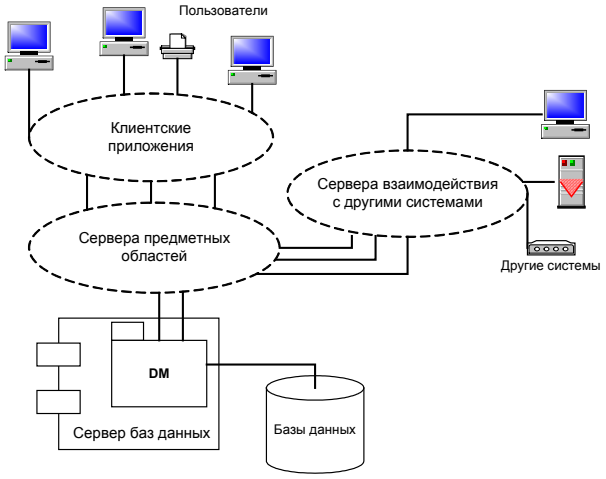

Рис. 4. Общий шаблон диаграммы пакетов
информационной системы корпоративного уровня

Этот шаблон применим при разработке приложения масштаба предприятия. Необходимость формулировки проекта программной системы в таком виде может возникать на этапах развития общего проекта автоматизации (при разработке приложений для второй, третьей и т.д. предметных областей). На этапе проектирования и реализации первого приложения серии достаточно воспользоваться упрощенным шаблоном (рис. 5).

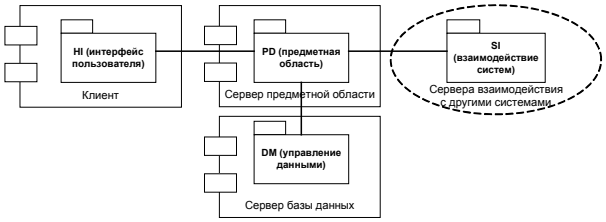

Рис. 5. Стартовый шаблон пакетов
информационной системы корпоративного уровня

Итак, в работе рассмотрен начальный этап проектирования информационной системы корпоративного уровня – создание пакетов приложения как семантически связанных групп объектов создаваемой программной системы.

ЛИТЕРАТУРА

1. *Gamma et al*. «Design Patterns». Reading, Massachusetts: Addison-Wesley, 1995.
2. *Коуд П., Норт Д., Мейфилд М.* Объектные модели. Стратегии, шаблоны и приложения. М.: Лори, 1999. 434 с.

Статья представлена факультетом математики и информатики Анжеро-Судженского филиала Кемеровского государственного университета, поступила в научную редакцию номера 3 декабря 2001 г.